THE EPA MULTIMEDIA INTEGRATED MODELING SYSTEM SOFTWARE SUITE

By Steven S. Fine, Program Manager, National Oceanic and Atmospheric Administration on Assignment to the U.S. Environmental Protection Agency, c/o U.S. EPA, MD E243-04, Research Triangle Park, NC, 27711, 919-541-0757, fine.steven@epa.gov; Steven C. Howard, Computer Specialist, National Oceanic and Atmospheric Administration on Assignment to the U.S. Environmental Protection Agency, Research Triangle Park, NC; Alison M. Eyth, Software Engineer, MCNC Environmental Modeling Center, Research Triangle Park, NC; Dean A. Herington, Research Assistant, Department of Computer Science, University of North Carolina, Chapel Hill, NC; Karl J. Castleton, Software Engineer, Pacific Northwest National Laboratory (formerly with U.S. EPA), WA

INTRODUCTION

Several trends in environmental modeling are driving a significant increase in the complexity of environmental modeling studies. These include the growing importance of

- combining models from multiple physical media (e.g., air, water, soil) or disciplines to make predictions that include an increasingly complete set of processes and outcomes;
- performing sensitivity and uncertainty studies to understand factors that affect results and to estimate the confidence that should be associated with predictions; and
- comparing multiple models and data sets that are intended to represent similar processes or contain similar information to understand models' and data sets' biases and errors.

Those and related challenges affect a number of projects at the U.S. Environmental Protection Agency (EPA). Examples include studying cross-media pollution or nutrient transport and transformation, tracking pollutants from their source to human doses via multiple pathways, predicting the relationships between climate and air quality at regional and global scales, and understanding risks associated with hazardous waste. If approaches typically used for executing and evaluating individual models are applied to such complex problems, a prohibitive amount of effort could be required and there is a significant probability that configuration and operational mistakes will corrupt the results.

To help modelers manage such increasingly complex simulations, a number of groups, including the EPA, have developed software systems that support modeling (e.g., Rizzoli and Young, 1997; Dennis *et al*., 1996; Leavesely *et al*., 1996; Laniak, 1999). These systems typically provide tools, software libraries, and/or software environments that simplify or partially automate common operations, including composing, configuring, executing, and evaluating models; but they lack the combination of power and flexibility required to effectively support some types of complex simulations. Issues that require further attention include feedbacks between models; suitability for models with different spatial and temporal scales; a conceptual design that cleanly supports interchanging models and data sets and modeling of physical, chemical, biological, and human systems; cross-platform portability; support for off-the-shelf models; and distributed computing.

MULTIMEDIA INTEGRATED MODELING SYSTEM SOFTWARE SUITE

We are developing the Multimedia Integrated Modeling System (MIMS) software suite to address the EPA's current and future interdisciplinary modeling needs. The MIMS software will allow modelers, including model developers and risk assessors, to conduct complex studies with less effort and greater confidence that the results represent the modelers' intent. MIMS will not make scientific or management decisions for modeling studies, but MIMS should allow modelers to better focus their attention on those issues.

The MIMS software suite will support the following activities:

- Data management and manipulation
- Model coupling and swapping
- Repetitive work, such as modeling multiple locations and conducting sensitivity and uncertainty studies
- Computation management, such as managing model executions on remote computers
- Model evaluation

The software suite will support these activities by automating common actions, simplifying manual operations, checking consistency based on information provided by modelers, and providing tools for various activities.

To effectively meet the needs of users, we have identified a number of desirable characteristics for the software suite. The suite should:

- Be applicable to a wide variety of environmental issues;
- Be compatible with multiple families of models and multiple sources of data (e.g., field observations, satellites, models);
- Support feedback between models;
- Be easy for modelers to use;
- Run on multiple operating systems (e.g., Windows, UNIX, Linux);
- Support distributed computing in a flexible manner;
- Be open source;
- Allow modelers to incorporate their models into the system without losing "ownership";
- Allow users to choose their level of investment (e.g., achieve some benefit with limited effort or achieve greater benefit with greater effort); and
- Make common operations easy to perform while also supporting less common and more complex operations.

The MIMS software suite will include several components, as shown in Figure 1. The "framework" provides a software infrastructure for composing, executing, and evaluating models. MIMS tools provide functionality required to prepare models or their inputs or to operate on models' results. In the future, MIMS software libraries will provide a standard implementation of common functions required by models. For instance, MIMS might provide a standard protocol (preferably based on off-the-shelf software) for representing and interchanging
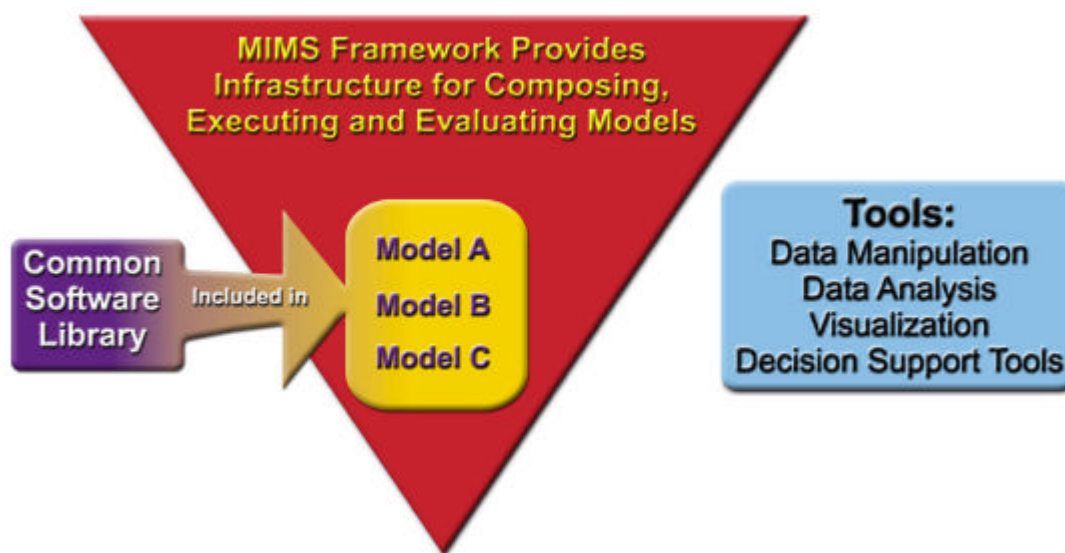
Figure 1: Components of the MIMS software suite.

environmental data which modelers could choose to use. There are currently no MIMS libraries. The MIMS framework and tools are described in the following sections.

MIMS Framework

Composing and Interchanging Models and Data Sets: To provide a flexible and robust method of combining and interchanging models and data sets, MIMS uses a modified version of the modeling paradigm from Argonne National Laboratory's Dynamic Information Architecture System (DIAS) (Christiansen, 2000) and the supporting DIAS software library. In the DIAS paradigm, one or more modelers decompose a system to be modeled into "domain objects" that represent the important things or concepts in the simulation. Examples of domain objects include an aquifer, a pollutant source, the atmosphere, homes, and a fish population. Each domain object contains parameters and processes. Parameters are attributes that describe the domain object, and processes are behaviors the domain object exhibits. Models provide the implementation of processes that are active in a simulation, as shown in Figure 2. Each model is defined to read data from and write data to a domain object's parameters, as shown in Figure 3. In essence, each domain object's parameters serve as a standard for any information about that domain object. For example, in Figure 3 Domain Object A could represent a lake. Its parameters might include depths, temperatures, and nitrogen concentrations. Domain Object B could represent an urban region with parameters of population, sewage treatment type, and economic activity. Model B could implement a discharge process by computing the amount of nutrients the urban region contributes to the lake. Model A could implement an aquatic chemistry process by computing the lake's nutrient concentrations. Since each model is defined in terms of the data standards provided by the domain objects, models conceptually do not interact directly with other models. This allows a modeler to replace a model with another implementation of the associated process without affecting other models in a system. This also allows a modeler to easily remove a model from the simulation and to instead incorporate data sets that contain the same type of information that the model would produce. Data analysis and model evaluation programs can also be included in scenarios as "models."
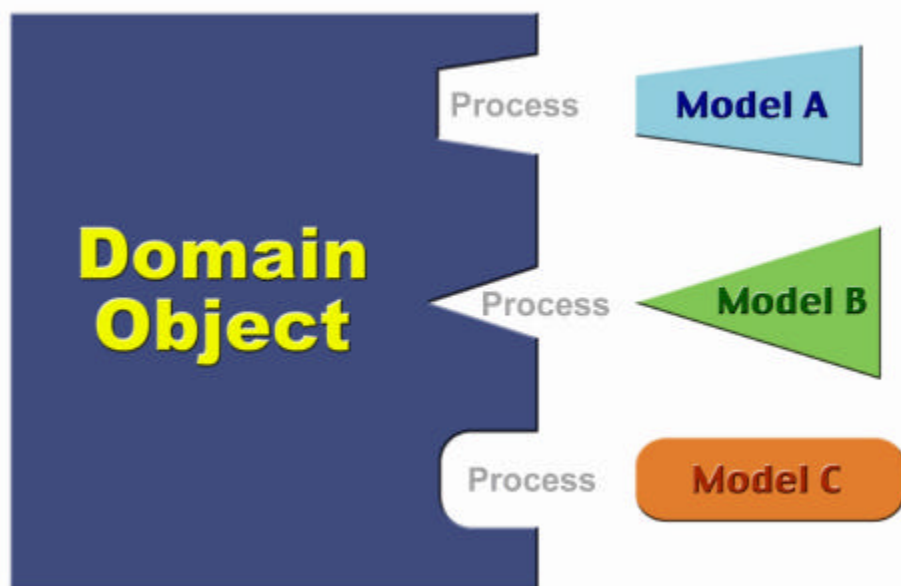
Figure 2: Models provide the implementation of the domain object's processes that are active in a simulation.

The primary difference between the MIMS and DIAS model coupling paradigms is that DIAS includes another layer of software between a model and a domain object which translates between the model's and the domain object's parameters and assumptions. If that functionality is required in MIMS, a wrapper is placed around an existing model and the combination of wrapper and existing model is treated by MIMS as a model. The MIMS approach allows a layer of code to be eliminated for models that do not require translation with little or no loss of flexibility.

To support a wide range of modeling issues, MIMS uses a very general concept of model parameters. Parameters provide the attributes of domain objects and the inputs and outputs of models. MIMS parameters check their consistency with other parameters and can provide customized user interfaces for editing. A parameter's developer determines the scope of its consistency checks. The basic MIMS parameters primarily check for compatible types of information (e.g., confirm that a floating-point number is provided when one is required). MIMS places few other requirements on parameters, which allows parameters to represent a broad range of information. MIMS already provides parameters for basic data types, such as floating-point numbers, strings, and integers, as well as more complex data structures, such as files, sets of chemical reactions, and descriptions of regular grids. Java code for new parameters can be easily incorporated into MIMS. Some parameter types provide an abstract description of data that models can use without being tied to a specific data source. For instance, models can be defined to read a time series of values without making any assumptions about where the data are stored (e.g., in a file, in memory, in a database). The MIMS approach for parameters is based on some concepts used in the computer framework developed for the Total Risk Integrated Methodology project (Palma *et al.*, 1999) with significant extensions.

The effort required to use a new model in MIMS varies greatly. If the model is not consistent with the process and parameter standards set by the domain objects, then the model must be adapted or a wrapper placed around the model. A wrapper's responsibilities might include converting units, computing derived parameters, and interpolating data. For all models, MIMS

4

Figure 3: Models are defined to read from and write to domain objects.

requires some basic information including the type of domain object to which the model applies, the process the model implements, how the model should be invoked, and the model's input and output parameters. A user can define this information in a graphical user interface, a text file, or Java code. Modelers can define their own models or use definitions someone else prepared. MIMS provides several features that make it possible to use some existing ("legacy") models in MIMS without having to write any new code. The person who defines a legacy model in MIMS can specify that input parameters should be passed to the model on the command line, via environment variables, in a textual control file, and/or as columnar time series data. MIMS will create the required control and time series files before invoking the model. MIMS can also read model outputs from columnar time series files and make those outputs available to other models in a MIMS scenario.

Users bring together or compose domain objects and models to create modeling scenarios. MIMS uses the information about domain objects and models to perform some basic consistency checks as the scenario is created. The consistency checks include verifying that all required input parameters are defined and that models' input and output parameter types and units are consistent with those of the domain objects.

MIMS also allows scenarios to be composed. A scenario, which might invoke a number of models, can be defined as a new type of model and attached to a domain object's process in another scenario. This is similar to the concept of wrapping Fortran statements in a subroutine and provides the same benefits of conceptual encapsulation and reuse of processing instructions. For example, a user could create a scenario that simulated aquatic biology via models of algae, fish, etc. Then that scenario and the multiple models it contains could be used to implement an aquatic biology process in a scenario that simulated multimedia nutrient transport.

Executing Scenarios and Models: After a user defines a new modeling scenario or opens a predefined scenario, he executes the scenario. MIMS passes descriptions of the scenario's

domain objects and models to the DIAS library, which invokes the models in the proper sequence. DIAS invokes a model when its input parameters are available and have changed since the model's last invocation.

We are currently developing capabilities to distribute model executions to remote computers. This will allow users to easily utilize remote compute servers from their desktops to significantly decrease the turnaround times for some types of simulations. For instance, Monte Carlo uncertainty analysis can be performed much more quickly if multiple computers are used. Our design includes the use of multiple protocols for working with remote computers so MIMS can be compatible with a variety of networking, computer architecture, software, and security environments.

Iteration: Repetitively running a set of models for different inputs is a very common activity. Examples of iterative studies include sensitivity and uncertainty studies, model calibration, optimization, and model execution for different sites, time periods, or management assumptions.

MIMS includes a general iterator design that is flexible and easy to extend. The design allows iterator developers to easily take advantage of MIMS's user interfaces, parameter consistency checking, and execution management while specifying whether realizations should be executed sequentially (e.g., for optimization) or in parallel (e.g., for multiple sites) and what post-processing of results should be performed (e.g., to compute the importance of factors during a sensitivity study). MIMS already includes a basic Monte Carlo iterator, and collaborators are developing an extensive uncertainty package that will plug into MIMS. MIMS iterators can be applied to any MIMS model, including complex scenarios that invoke multiple legacy models.

MIMS Tools: Typically, modelers invest significant effort preparing models and their inputs and analyzing and utilizing model results. MIMS tools assist with the most common of those activities. One tool is currently available and another two tools are under development.

Modular Spatial Allocator: When dealing with spatially explicit data sets, a very common operation is to allocate attributes from one set of polygons, lines, or points to another set of polygons, lines, or points. For instance, attributes that are specified by county (e.g., atmospheric emissions of pollutants) might be allocated to model grid cells, or fluxes on a fine mesh might be aggregated to a coarse mesh. This is a standard operation in a geographic information system (GIS), but some communities that use MIMS would benefit from a stand-alone spatial allocator because they do not have expertise with GIS packages, because they have little or no additional need for a GIS, or because it can be difficult or inefficient to invoke a GIS as part of an interface between two models.

To address these concerns, MIMS includes a modular spatial allocator. This application reads a source set of polygons, lines, and points with an associated attribute and a destination set of shapes, computes the overlaps of the source and destination shapes, allocates the input attribute, and writes the results. The spatial allocator has been designed in a manner that makes it relatively easy to add additional input and output formats for data or to add a different spatial allocation algorithm.

Plotting Tool: Many users would like to plot model results and comparisons in scatterplots, time series plots, histograms, bar charts, and boxplots (box-and-whisker plots). Commercial packages can perform many of these functions but are often platform-specific, may not be amenable to operation in a batch mode where no graphical user interface is present, and cannot be distributed as part of a free modeling system.

MIMS already allows users to include scatterplots, time series plots, and histograms in their scenarios. We are also designing a more general and powerful plotting facility that will combine the best features of framework-embedded plotting facilities and a stand-alone, easy to use plotting application. The goal is to allow users to use the same data analysis tool to easily analyze their data outside of the framework, to create templates containing plot compositions and formats to use in their scenarios, and to automatically create plots on-screen or in a publication-quality file format when they execute their MIMS scenarios. To achieve this we are selecting an existing plotting application that is open source, portable across platforms, extensible, and easy to use and that produces a wide variety of plots on-screen and in publication-quality formats. We will then add features for bidirectional communication with the framework. The plotting application will create plot templates that the framework can use and the framework will invoke the plotting application with the data the user would like to plot.

Thematic Mapper: We are also pursuing a similar approach to providing a thematic mapping capability in MIMS. Such a tool could provide an easy way to quickly view model inputs or results in a geospatial context. We are evaluating open source, cross-platform mapping components or applications, such as OpenMap, that could be invoked from within MIMS as well as operate in a stand-alone manner.

<u>Development Approach</u>: To try to maximize the value to customers and the timeliness of our work, we have adopted some approaches from the Extreme Programming software development methodology (Beck, 1999) while tempering those approaches to account for the special requirements of an extensible framework, our very small and scattered development team, and the inertia of a large organization. While we have a long-term vision for MIMS, we typically select short-term priorities based on customers' needs and schedules. When considering one customer's needs we also consider if there is a more general issue that is relevant to multiple customers. We are also collaborating with other agencies that are developing modeling tools to try to identify and share standard subsystems.

APPLICATIONS OF MIMS

EPA's Council for Regulatory Environmental Modeling (CREM) encourages the use of common best practices in EPA's modeling groups. CREM is expecting MIMS to provide a platform for groups that require modeling frameworks to conduct complex studies.

To help us achieve the generality required to meet CREM's expectations, we are working with groups that have very different types of models and applications. Working with these multiple groups helps ensure that the MIMS design is general, provides a broad evaluation of MIMS approaches, and helps guide growth in a variety of directions. The projects that currently plan to use MIMS include the following:

- Total Risk Integrated Methodology (TRIM). TRIM will support risk assessment for hazardous pollutants that are emitted to the air and then are transported to soil and water, such as mercury. MIMS will provide the TRIM project a platform for coupling the models required for their risk studies and data analysis tools.
- Community Multiscale Air Quality (CMAQ) model. CMAQ is a state-of-the-art grid-based air quality model. MIMS will provide a graphical alternative for configuring CMAQ and for managing repetitive model executions.
- Clean Air Status and Trends Network (CASTNET). MIMS will provide a graphical user interface, simulations of multiple sites and years, and data analysis tools for the application of the Multilayer Model of dry deposition (Meyers *et al*., 1998) to CASTNET data.
- New Generation Compartmental Model. This project uses MIMS as a platform for exploring new approaches for constructing fully integrated compartmental models of multiple media, including biota.
- Urban Drainage Decision Support System. An external group supported by a cooperative agreement funded by EPA's Office of Water is using MIMS as the basis for a prototype decision support tool for urban drainage applications. This includes the development of uncertainty analysis and optimization tools within MIMS.

While those projects will be using the MIMS software suite, they are not currently designing their domain objects or models to be interoperable. A few of the projects have expressed interest in using another project's models with MIMS's support in the future, but they have not invested any effort yet in the conceptual design and model adaptations that will be required to achieve that interoperability.

FUTURE DIRECTIONS

During the next couple of years, new MIMS capabilities will most likely address two issues: very large computations and working with environmental data. More detailed observations and representations of environmental processes, increased interest in long-duration environmental simulations, and growing demand for sensitivity and uncertainty estimates are significantly increasing both the computational resources required to perform an individual simulation and the number of simulations. Several planned MIMS capabilities will support such computationally intensive work. As described above, we are designing distributed computing support for MIMS that will allow a user to easily utilize remote compute servers from her desk. In the future MIMS will be extended so the computation management portion of MIMS can continue running on a server even when a user turns off the machine where she started MIMS. This will enable MIMS to manage simulations that require weeks or more of time without being tied to a desktop machine. Also, we are considering adding a script language to MIMS to provide another avenue for automating computations.

We are starting to shift the emphasis of our development from computation management to all facets of working with environmental data. We will provide in MIMS common data reduction tools, such as computing averages, subsets, and extrema. After we have developed the basic plotting and thematic mapping tools described above, we will investigate off-the-shelf applications that can provide an integrated view of three-dimensional, time varying data sets that are not on the same grid or mesh, such as overlapping results from models at two different

resolutions. We will also investigate data representation and interchange approaches that can foster sharing data among independently developed models and data manipulation and analysis tools. This may apply or build on existing approaches such as the Synthetic Environment Data Representation and Interchange Specification (Foley *et al.*), the Distributed Oceanographic Data System (Unidata, 2002), and the Earth Science Markup Language (Ramachandran, 2001). Another issue we expect to address is how to track model results and find data sets that might be located at multiple institutions.

SUMMARY

The MIMS software suite will allow modelers to focus more on scientific and policy issues while conducting increasingly complex modeling studies. The suite supports composing, configuring, executing, and evaluating a wide range of models. We are working with diverse modeling projects to identify and support their common requirements and to evaluate our success towards that goal. The current status of MIMS is available at http://www.epa.gov/asmdnerl/mims.

REFERENCES

Beck, K., 1999, *Extreme Programming Explained: Embrace Change*, Addison-Wesley.

Christiansen, J. H., 2000, A Flexible Object-Based Software Framework for Modeling Complex Systems with Interacting Natural and Societal Processes. *Proceedings, 4th International Conference on Integrating GIS and Environmental Modeling (GIS/EM4): Problems, Prospects and Research Needs. Banff, Alberta, Canada, September 2 – 8.*

Dennis, R. L., Byun, D. W., Novak, J. H., Galluppi, K. J., Coats, C. J., Vouk, M. A., 1996, The Next Generation of Integrated Air Quality Modeling: EPA's Models-3. *Atmos. Environ.*, **30**, 1925-1938.

Foley, P. G., Mamaghani, F., Birkel, P. A., 1998, The Synthetic Environment Data Representation and Interchange Specification (SEDRIS) Development Project. http://www.sedris.org/pr11trpl.htm.

Laniak, G. F., 1999, Documentation for the FRAMES-HWIR Technology Software System, Volume 1: System Overview, http://www.epa.gov/epaoswer/hazwaste/id/hwirwste/pdf/risk/system/s0499.pdf.

Leavesley, G.H., Markstrom, S.L., Brewer, M.S., and Viger, R.J., 1996, The Modular Modeling System (MMS) - The Physical Process Modeling Component of a Database-Centered Decision Support System for Water and Power Management. *Water, Air, and Soil Poll.*, **90**, 303-311.

Meyers, T.P., Finkelstein, P., Clarke, J., Ellestad, T. G., Sims, P. F., 1998, Description and Evaluation of a Multilayer Model for Inferring Dry Deposition Using Standard Meteorological Measurements. *J. Geophys. Res.*, **103(D7)**, 22,645-22,661.

Palma, T., Vasu, A. B., Hetes, R. B., 1999, The Total Risk Integrated Methodology (TRIM). *Environ. Manager*, **5**, March, 30-34.

Ramachandran, R. Alshayeb, R. M., Beaumont, B., Conover, H., Graves, S., Li, X., Movva, S., McDowell, A., Smith, M., 2001, Earth Science Markup Language: A Solution for Generic Access to Heterogeneous Data Sets. http://esml.itsc.uah.edu/presentations2.html.

Rizzoli, A. E., Young, W. J., 1997, Delivering Environmental Decision Support Systems: Software Tools and Techniques. *Environ. Modelling & Software*, **12**, 237-249.

Unidata, 2002, Distributed Oceanographic Data System Web Site, http://www.unidata.ucar.edu/packages/dods/index.html.